

# Estructura de un Proyecto Android

## 1. AndroidManifest.xml (Archivo de configuración principal)

xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ejemplo.miapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/Theme.MiApp">

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

¿Qué hace?

- Define el nombre del paquete de tu app
- Lista todas las Activities (pantallas)
- Define cuál es la Activity principal (LAUNCHER)
- Declara permisos si los necesitas

## 2. Layout XML (res/layout/activity\_main.xml)

xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp"
        android:gravity="center">

<!-- Etiqueta -->
<TextView
    android:id="@+id/tvTitulo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Introduce tu nombre:"
    android:textSize="18sp"
    android:layout_marginBottom="16dp"/>

<!-- Caja de texto -->
<EditText
    android:id="@+id/etNombre"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Escribe aquí"
    android:inputType="textPersonName"
    android:layout_marginBottom="16dp"/>

<!-- Botón -->
<Button
    android:id="@+id/btnSaludar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Saludar"/>

</LinearLayout>

```

#### Conceptos clave del Layout:

- **android:id**: Identificador único para acceder desde el código
- **layout\_width/height**:
  - **match\_parent** = ocupa todo el ancho/alto disponible
  - **wrap\_content** = ajusta al contenido
- **orientation="vertical"**: Los elementos se apilan verticalmente
- **padding**: Espacio interno (dp = density-independent pixels)
- **gravity**: Alineación de los elementos dentro del contenedor

### 3. Activity (Código Java) (MainActivity.java)

```
java
package com.ejemplo.miapp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    // Declaración de variables para los componentes
    private EditText etNombre;
    private Button btnSaludar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Vincula esta Activity con el layout XML
        setContentView(R.layout.activity_main);

        // Conectar las variables con los componentes del layout
        etNombre = findViewById(R.id.etNombre);
        btnSaludar = findViewById(R.id.btnSaludar);

        // Configurar el evento click del botón
        btnSaludar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Obtener el texto del EditText
                String nombre = etNombre.getText().toString();

                // Validar que no esté vacío
                if (!nombre.isEmpty()) {
                    // Crear el mensaje
                    String mensaje = "Hola " + nombre;
                }
            }
        });
    }
}
```

```

        // Mostrar Toast (mensaje superpuesto)
        Toast.makeText(MainActivity.this, mensaje,
Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(MainActivity.this, "Por favor introduce tu
nombre", Toast.LENGTH_SHORT).show();
    }
}
});
}
}
}

```

¿Qué hace cada parte?

1. **onCreate()**: Método que se ejecuta al crear la Activity
2. **setContentView(R.layout.activity\_main)**: Carga el layout XML
3. **findViewById()**: Busca un componente por su ID
4. **setOnClickListener()**: Define qué hacer cuando se pulsa el botón
5. **getText().toString()**: Obtiene el texto del EditText
6. **Toast.makeText()**: Crea el mensaje emergente
  - LENGTH\_SHORT: Duración corta (2 segundos)
  - LENGTH\_LONG: Duración larga (3.5 segundos)

#### 4. Clase R (Generada automáticamente)

```

java
R.layout.activity_main // Referencia al archivo XML
R.id.etNombre          // Referencia al ID del EditText
R.id.btnSaludar        // Referencia al ID del Button

```

¿Qué es R?

- Clase generada por Android Studio automáticamente
- Contiene referencias a todos los recursos (layouts, IDs, strings, colores, etc.)
- **Nunca la edites manualmente**

#### 5. Strings.xml (res/values/strings.xml) - Opcional pero recomendado

```

xml
<resources>
    <string name="app_name">Mi App</string>
    <string name="titulo">Introduce tu nombre:</string>
    <string name="hint_nombre">Escribe aquí</string>

```

```
    <string name="boton_saludar">Saludar</string>
</resources>
```

**Buena práctica:** En lugar de poner textos directamente en el XML, usa referencias:

```
xml
android:text="@string/titulo"
...

---
```

## ## Flujo de Ejecución

1. **\*\*Al abrir la app:\*\***
    - Android lee el `AndroidManifest.xml`
    - Encuentra que `MainActivity` es la Activity principal
    - Ejecuta `onCreate()` de `MainActivity`
  2. **\*\*En onCreate():\*\***
    - `setContentView()` infla (carga) el layout XML
    - `findViewById()` conecta las variables Java con los componentes visuales
  3. **\*\*Cuando el usuario pulsa el botón:\*\***
    - Se ejecuta el código dentro de `onClick()`
    - Se obtiene el texto del `EditText`
    - Se muestra el `Toast` con el mensaje
- 

## ## Estructura de Carpetas

```
...
MiApp/
├─ app/
│   └─ src/
│       └─ main/
│           └─ java/
│               └─ com/ejemplo/miapp/
│                   └─ MainActivity.java
│           └─ res/
│               └─ layout/
```

```
| | | | | └─ activity_main.xml
| | | | └─ values/
| | | | └─ strings.xml
| | | └─ mipmap/
| | └─ ic_launcher.png
| └─ AndroidManifest.xml
└─ build.gradle (configuración de la app)
└─ build.gradle (configuración del proyecto)
```

## Conceptos Clave para Recordar

1. **Activity** = Una pantalla de tu app
2. **Layout XML** = Define CÓMO se ve la pantalla
3. **Código Java** = Define QUÉ hace la app (lógica)
4. **R.java** = Puente entre código y recursos XML
5. **Toast** = Mensaje emergente temporal